

实验一 词法分析与语法分析

组长：周心萌 141220161 ddzhouxm@163.com
组员：银琦 141220132 141220132@smail.nju.edu.cn

1 实现功能

1.1 功能介绍

编写一个程序对使用 C++ 语言书写的源代码进行词法分析和语法分析，并打印分析结果。程序要求能够查出词法错误，语法错误，除此之外要能够识别八进制和十六进制数。

1.2 实现过程

1.2.1 词法分析

按照 C 语言语法手册定义了文法，额外增加了对八进制及十六进制的判断，同时增加了两种情况：错误的八进制和错误的十六进制，便于遇到输入错误的八进制和十六进制时可以向用户说明。

```
42 OCT [0][0-7]+
43 WRONGOCT [0][0-7]*[89]+
44 HEX [0][xX][0-9a-fA-F]+
45 WRONGHEX [0][xX][0-9a-fA-F]*[g-zA-Z]+
```

1.2.2 语法分析

根据 C 语言手册定义合格的语法，并根据相应的错误对其进行错误恢复，如在语句中发生错误即使用 error SEMI，将句末的分号作为开始恢复的 token，从而实现正确的错误恢复。同样，函数定义时若发生错误，则使用函数的右括号进行错误恢复

```
| error SEMI {IFError=1;}
| error WHILE LP Exp RP {IFError=1;}
| error IF LP Exp RP {IFError=1;}
```

在进行语法分析检测的同时，建立语法树 (create(char *type, int count, ...)) 语法树的结构包括：节点内容，子女节点，节点所在行数，节点名称，节点标记（标记是否是正确的词法）。

在创建语法树时，使用不定参数形式，采用 va_list 读入参数。

```
va_list n;
va_start (n, count);
```

参数创建为对应节点的子女节点，
返回该节点

1.2.3 打印语法树 (print(struct treeNode* root, int depth))

将所建立的语法树从顶部开始输出，将子女从左到右输出，并重新调用 print 函数进行以子女为根的语法树进行输出。

1.3 完成情况

可以对测试文件进行词法分析及语法分析，可以识别出所有的词法及语法错误，能够识别出测试文件中的八进制和十六进制数，并且能够判断其正确性。

2 编译环境及输入格式

2.1 编译环境

Ubuntu 12.04 flex 2.5.35 bison 2.5

2.2 输入格式

进入 Code 文件夹，输入 make，对于在 Code 外 Test 文件夹中的测试文件，输入 make test；对于在 Code 内的测试文件，输入 ./ parser filename。

3 遇到的问题

- 3.1 在写词法分析的规则部分时，没有注意处理顺序，一开始将 ID 写到了关键字的前面，在词法分析阶段没有问题，但是语法分析时不能够识别出正确的类型。
- 3.2 在语法分析阶段没有对错误的八进制和十六进制进行处理，导致同时输出了词法错误和语法错误。
- 3.3 在写错误恢复时，错误的将实验讲义上的所有错误恢复输入(Exp : error RP)，导致在测试时进行了错误的恢复，不能识别出两个错误。
- 3.4 在输出语法分析树的时候，将 print 函数写在了 yystart()前面，还没有建立语法树就将语法树进行输出，致使 print 什么都没有输出。
- 3.5 判断分号缺失的错误时，错误行数移至下一行（该问题可以接受），但是下一行的错误无法被正确判断，后在 speicier 中加入 error 修复该错误